Charlie's RFID Teddy Bear

Written By: David Harris

TOOLS:

- Computer (1)
- Marking pen (1)
- Phillips screwdriver (1)
- SD memory card adapter (1)
  if your computer doesn't have a built-in SD slot
- Scissors (1)
- Sewing needle (1)
- Solder (1)
- Soldering rion (1)
- Tape (1)
- USB programming cable (1)
- Wire cutter/stripper (1)

PARTS:

- Charlie Bear Project Bundle (1)
- Stuffed toy (1)
  Choose something quite soft that can accommodate the electronics easily and also provide some padding for them.
- Arduino microcontroller (1)
- Wave Shield (1)
  an audio shield for Arduino
- Parallax RFID Reader (1)
  aka RFID antenna
- RFID tags (1)
  Parallax offers several different shapes/sizes.
- Speaker, 3" (1)
- Connector header (1)
  0.1" spacing, with 4 jumper wires
- Wire (1)
  multiple colors
- Velcro (1)
  that matches toy color

- Thread (1)
  that matches toy color
- Barrel connector (1)
  power plug for Arduino
- Battery holder, 6xAA (1)
- Batteries, AA (6)
- SD memory card (1)
- Battery clip (1)
  if battery holder has 9V connector rather than wire leads

SUMMARY

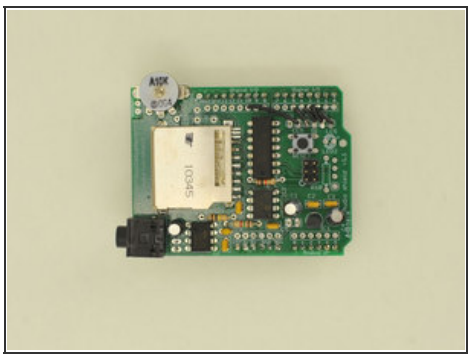
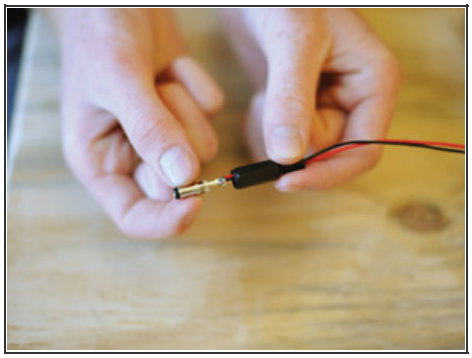
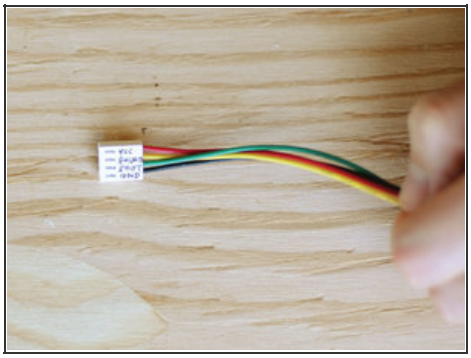
This location-aware teddy bear reads RFID tags and plays different customizable sounds depending on where it is or what object it's near — other toys, books, CDs, anything.

Charlie's Bear helps children explore the world around them by producing sounds in reaction to other toys or objects nearby. It can play any sound files you upload to the memory card inside — for example, the voice of the bear, a noise that another toy might make, a theme song prompted by a toy from a TV show, or a reading of a favorite book.

I created the toy for my nephew Charlie, who was born with cerebral palsy. Charlie's vision is poor, but he's very tactile and auditory. This toy takes advantage of his excellent hearing and the joy he derives from music and sounds. And for all young children, this toy is an easy and safe way for them to pick their own music. Just bring a CD case (or other tagged item) near, and the bear plays it — no complicated CD player or computer.

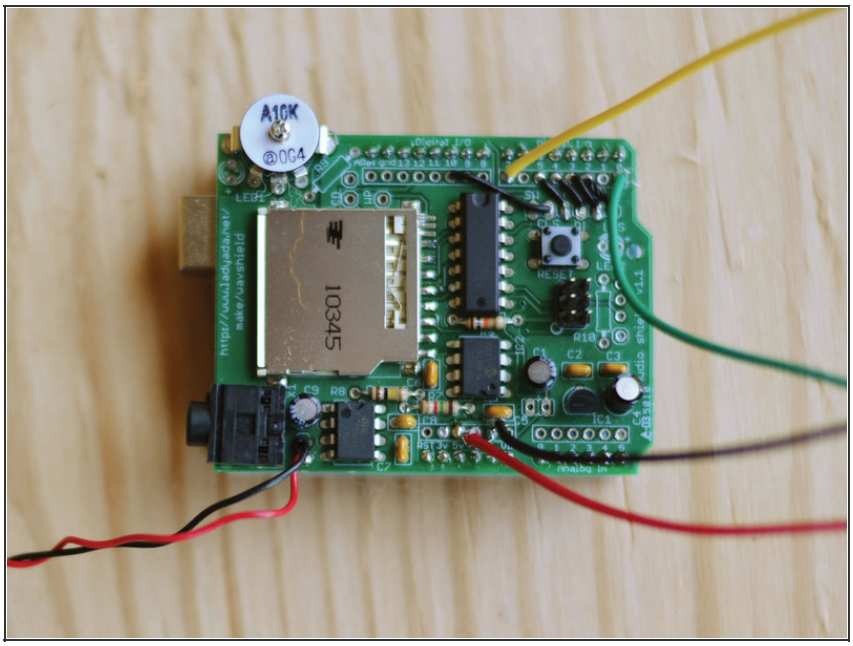
At the heart of Charlie's Bear, an Arduino microcontroller uses a radio frequency identification (RFID) reader to recognize nearby RFID tags, then uses an audio shield attachment to play corresponding audio files stored on an SD memory card. The SD card stores about a minute of audio content per megabyte, so a cheapo 4GB card will hold more than 60 hours.

Step 1 — Assemble the electronics.

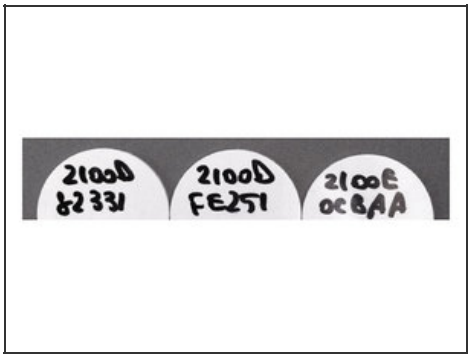
- Solder the Wave Shield together, following the instructions at http://ladyada.net/make/waveshield. At the step where you screw the plastic volume dial onto the housing, just insert the screw without the plastic dial, and turn it to maximum volume level.
- Cut and strip 4 wires about 6"–8" long, and solder them to the 4-pin connector, leaving the other ends bare. Mark the connector positions Vcc, Enable, SOut, and GND, to match the RFID reader's serial header, and connect red and black wires to voltage and ground, respectively.
- If your battery holder has wire leads, solder its red wire to the inner terminal of the barrel connector (power plug) and its black wire to the outer terminal. (If it has a 9V connector, solder the wires from a 9V battery snap.)
- Screw the housing onto the plug and wrap the wires with electrical tape if needed to hold things firm. Solder the wires to the 9V battery clip and insulate with more tape.
- The length of the wires isn't critical, but leave enough room to place the RFID sensor in the bear's chest and have the wires come out the back to reach the rest of the electronics.

Step 2



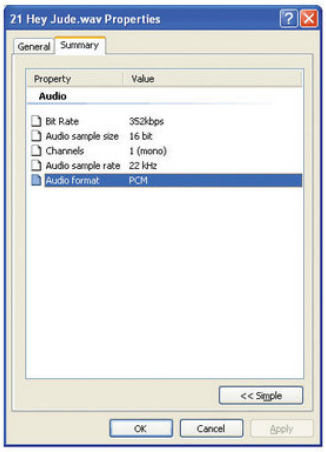
- Solder two 6" wires to the speaker if it doesn't already have leads connected.
- Solder the speaker wires to the 2 holes on the Wave Shield right next to capacitor C9, behind the headphone jack. Either wire can go in either hole.
- From your 4-wire RFID connector cable, attach the red and black wires to the Wave Shield's +5V and GND holes, respectively, and solder the SOut wire to digital I/O pin 0 and Enable wire to pin 7.
  - Vin -> +5v
  - Gnd -> Gnd
  - Enable -> Pin7
  - SOut -> pin0
- To complete the electronics, just plug the Wave Shield onto the Arduino, connect the 4-wire cable to the RFID reader (make sure it goes the right way around), and plug the battery power plug into the Arduino.

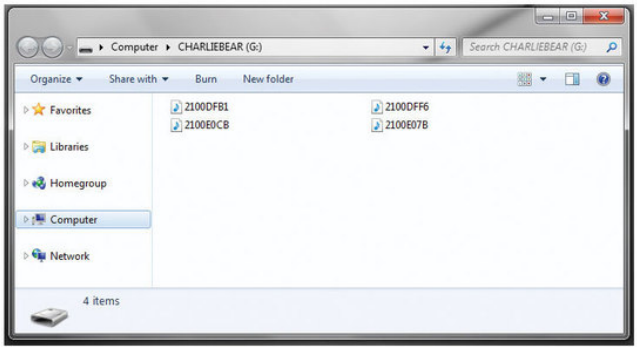
Step 3 — Determine the RFID tags' IDs.



- Install the Arduino environment if you haven't already, and download the Charlie Bear software, RFID tag identifier, and the Wave Shield Library. Unplug the RFID reader. Open *RFIDread.pde* and upload to the Arduino using a USB cable.
- Open the Serial Monitor to see what the RFID reader is seeing, then plug the RFID reader back in. Bring each one of your RFID tags close to the reader. The 10-digit hexadecimal tag IDs should print out in the Serial Monitor.
- Use a permanent marker to label each tag with its tag ID. You won't need to do this again until you get a new batch of tags.

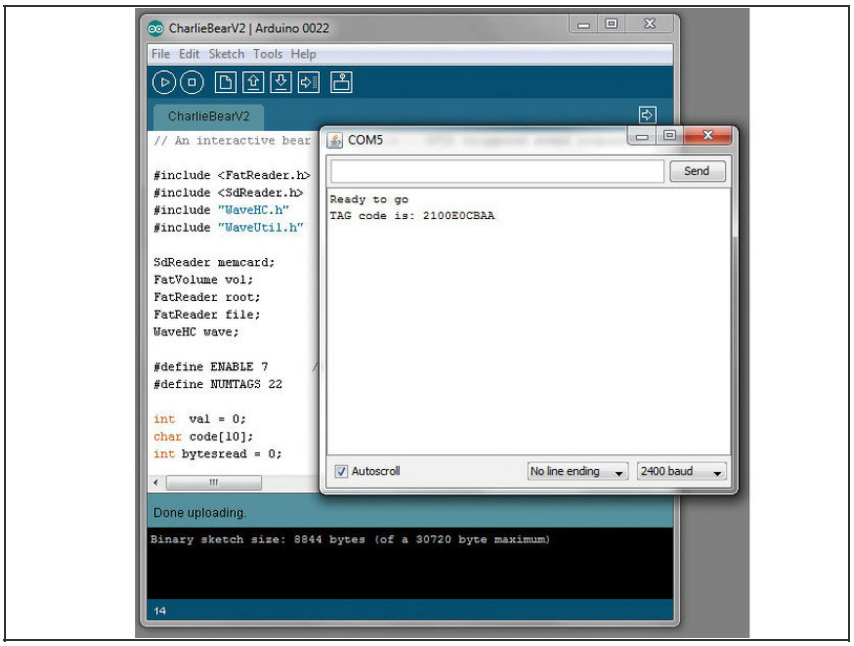
Step 4 — Encode the audio.





- For each tag, record your audio or otherwise obtain a sound file you want to use.
- Download and install Audacity — free, open source software for recording and editing sounds. Following the Wave Shield's "Converting audio to the proper format" tutorial, use Audacity to convert your audio files into the correct format: 16-bit sample size, PCM encoding, and a sample rate of 22kHz or less.These conversions might be the trickiest part of the whole project.
- Name each sound file with the first 8 hexadecimal digits of the RFID tag you want to associate with it. (It's extremely unlikely that you'll have duplicates.)
- Copy the sound files in the root directory of the SD memory card.
- Since these filenames don't say anything about the sounds they contain, be sure to note somewhere which sound goes with which tag.

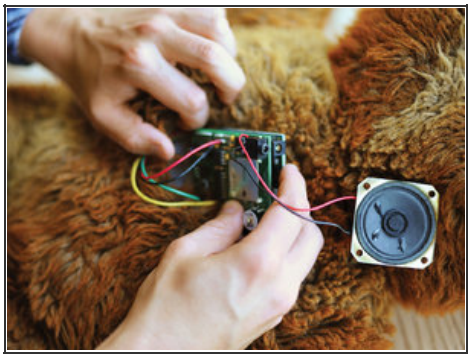
Step 5 — Configure and test the code.



- Unplug the RFID reader and load the *CharlieBear.pde* sketch into the Arduino.
- Plug the RFID reader back in, and see if bringing a tag near starts a sound playing.
- **Troubleshoot your code:** If there is no sound, check:
  - the power connections
  - that the RFID connector is oriented the right way
  - that your sound files are named correctly and are in the correct format.
- If you still experience problems, debug them by running the electronics while the Arduino is cabled to your computer, and watch the Serial Monitor for what the RFID reader sees and sends. For more help, see the comments at the top of the code.

Step 6 — Perform the plush toy surgical implant.





- Make a spinal incision in your bear or other plush toy large enough to get your electronics inside. How you perform this operation will depend on how the toy is made. For mine, I unstitched the bear along its backbone seam from just below the neck down to the waist.
- Attach a closure mechanism. I sewed velcro of a matching color inside one side of the incision and on the outside of the other. There was enough play in the bear's "skin" to allow the sides of the seam to overlap and the velcro to shut.
- There are many ways to reclose your bear, but you'll want to be able to get back inside, so don't just stitch it up. I recommend using velcro (not the adhesive kind). The main thing is to come up with something that a child can't open easily or by accident.
- Embed the electronics. Put the speaker behind the bear's mouth and nose. The RFID reader should go against the chest, and the rest of the electronics can go anywhere inside the middle of the bear. Take some stuffing out if it's hard to close the bear.
- If you know the stuffing isn't conductive (which it shouldn't be), you can just put the electronics straight in. But you may get fluff everywhere, so be careful when taking the memory card in and out. I've never encountered stuffing that conducts electricity, but if yours does, try enclosing your components in the anti-static bags they probably shipped in.
- Your toy is ready to play with!

Connecting Sounds With Objects

Combining object sensing with sound is more powerful than you might think. The most obvious application is just for the bear to make appropriate noises in response to other toys. But what about interacting with other objects, besides toys? An ID tag stuck inside the cover of a book can trigger the bear to read it. A tag in a CD case can trigger the bear to play the corresponding

music, without a small child's having to fuss with smudge- and scratch-prone CDs or age-inappropriate computers.

If you have a child who just won't listen to you, maybe they'll listen to their friend the bear. Record some tags in "bear voice" saying that it's time to go to sleep or time to brush our teeth. And don't forget to give the parents a keychain RFID tag that has no sound associated with it. This very handy tag will instantly make the bear go silent! These are some of the Charlie's Bear sound applications I've used successfully with the kids in my life, and I'm sure you can come up with many more.

Mod Charlie's Bear!

The software for Charlie's bear is fairly straightforward and easy to modify. For example, the existing code interrupts the current sound playing when a new tag is brought into range, but you can change its behavior so that it plays the current sound file until the end before starting the new one.

There are a few obvious physical modi-fications you might like to try. The bear has no power switch, so to turn it off you need to unplug the battery pack. Instead, you could add a switch inside a paw or combination of paws.

Alternatively, you might like to add some kind of motion-sensitive switch and modify the code so it uses less power when idle. With software power management, turning off the RFID antenna will save the most power. Then the batteries will last long enough that you might never need to switch your bear off.

Or maybe the bear would like to dance along with the sounds it makes? Consider adding a vibration motor or stepper motors that the Arduino can trigger along with the sound files. Perhaps you can modify the code so that some tags play sound and others cause vibration or movement.

Let us know how you use Charlie's Bear!

Resources

Charlie's Bear software — CharlieBear.pde from http://makeprojects.com/v/28RFID tag identifier — RFIDread.pde from http://makeprojects.com/v/28Arduino IDE — http://arduino.ccAudacity — http://audacity.sourceforge.net

This project first appeared in MAKE Volume 28, page 106.

This document was last generated on 2012-12-15 08:27:38 PM.